



A Parallel Full Geometric Multigrid Solver for Time Harmonic Maxwell Problems

Mathieu Chanaud, Luc Giraud, David Goudin, Jean-Jacques Pesqué, Jean Roman

► To cite this version:

Mathieu Chanaud, Luc Giraud, David Goudin, Jean-Jacques Pesqué, Jean Roman. A Parallel Full Geometric Multigrid Solver for Time Harmonic Maxwell Problems. [Research Report] RR-8335, INRIA. 2013, pp.25. hal-00847966

HAL Id: hal-00847966

<https://hal.inria.fr/hal-00847966>

Submitted on 24 Jul 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



A Parallel Full Geometric Multigrid Solver for Time Harmonic Maxwell Problems

M. Chanaud, L. Giraud, D. Goudin, J.J. Pesqué, J. Roman

**RESEARCH
REPORT**

N° 8335

July 2013

Project-Teams HiePACS



A Parallel Full Geometric Multigrid Solver for Time Harmonic Maxwell Problems

M. Chanaud*, L. Giraud*, D. Goudin[†], J.J. Pesqué[†], J. Roman*

Project-Teams HiePACS

Research Report n° 8335 — July 2013 — 25 pages

Abstract: The design of scalable parallel simulation codes for complex phenomena is challenging. For simulations that rely on PDE solution on complex 3D geometries, all the components ranging from the initialization phases to the ultimate linear system solution must be efficiently parallelized. In this paper we address the solution of 3D harmonic Maxwell's equations using a parallel geometric full multigrid scheme where only a coarse mesh, tied to geometry, has to be supplied by an external mesh generator. The electromagnetic problem is solved on a finer mesh that respects the discretization rules tied to wavelength. The mesh hierarchy is built in parallel using an automatic mesh refinement technique and parallel matrix-free calculations are performed to process all the finer meshes in the multigrid hierarchy. Only the linear system on the coarse mesh is solved by a parallel sparse direct solver. We illustrate the numerical robustness of the proposed scheme on large 3D complex geometries and assess the parallel scalability of the implementation on large problems with up to 1.3 billion unknowns on a few hundred cores.

Key-words: Geometric multigrid, preconditioning, 3D Maxwell/Helmholtz, parallel computing.

* Inria, CNRS (LaBRI UMR 5800) and Université de Bordeaux

[†] CEA/CESTA, CS60001, 33116 Le Barp, France

**RESEARCH CENTRE
BORDEAUX – SUD-OUEST**

200 avenue de la Vielle Tour
33405 Talence Cedex

Un solveur full-multigrille géométrique pour le problème de Maxwell harmonique

Résumé : Le développement de codes de calcul passant réellement à l'échelle pour la simulation de phénomènes complexes est un challenge. Pour des simulations basées sur des EDP définies sur des géométries complexes, toutes les composantes du calcul, qui vont souvent des phases d'initialisation à la phase ultime de résolution d'un système linéaire, doivent être parallélisées efficacement. Dans ce rapport, nous considérons la résolution des équations de Maxwell tridimensionnelles par un schéma multigrille géométrique pour lequel seul le maillage le plus grossier, qui capture correctement la géométrie, est fourni par un générateur de maillages externe au code. Le problème d'électromagnétisme est résolu sur un maillage plus fin qui satisfait les contraintes liant le pas de maillage et la fréquence étudiée. La hiérarchie de maillages est construite en parallèle via une technique de raffinement et des calculs sans matrice sont mis en œuvre sur l'ensemble des maillages plus fins dans la hiérarchie. Seul le système linéaire défini sur le maillage grossier est construit et résolu par une méthode directe parallèle. Nous illustrons la robustesse du solveur obtenu par la résolution de problèmes définis sur des géométries 3D complexes et démontrons le passage à l'échelle de l'implantation parallèle sur des problèmes de grande taille allant jusqu'à 1.3 milliards d'inconnues.

Mots-clés : Multigrille géométrique, préconditionnement, Maxwell/Helmholtz 3D, calcul parallèle.

Contents

1	Introduction	4
2	Some background on time harmonic Maxwell's equations	5
2.1	A strong coupling	6
2.2	A weak coupling	7
2.3	The design process for complex objects	8
3	The full geometric multigrid solver	8
3.1	Grid transfer operators	9
3.1.1	The prolongation operator $P_{\ell-1}^{\ell}$	9
3.1.2	The restriction operator $R_{\ell}^{\ell-1}$	11
3.2	Matrix-free smoothers	11
4	Numerical experiments	11
4.1	On the relaxation parameter of the Jacobi smoother	12
4.2	Influence of the wave number	12
4.3	MG as preconditioner	15
5	Parallel implementation and performance	17
5.1	Parallel Implementation	17
5.2	Parallel performances	19
6	Concluding remarks	22

1 Introduction

The design of stealthy objects requires the computation of Radar Cross Section (RCS) of the complex 3D targets with complex coatings. This implies solving numerically the Maxwell's equations with time harmonic hypothesis into penetrable bodies and the unbounded surrounding free space. The complexity of the coated targets and the required accuracy prevent the use of asymptotic methods of any type [8]. Discretization techniques such as finite element methods, for instance, need to be employed and several critical difficulties must be addressed.

The first difficulty is to handle the unbounded domain. An usual solution is to truncate the computational domain far enough from the target and use any kind of absorbing boundary condition on this truncating interface. Nowadays, the Perfectly Matched Layer technique introduced by Berenger [5] is the most often chosen solution. However for some applications, this solution is not accurate enough. For such simulations, the use of a Boundary Integral Equation (BIE) method on a surface surrounding the target object enables the full compliance with the radiation condition at infinity satisfied by the Maxwell's equations.

The second main difficulty is related to the size of the linear systems arising from the discretization of these equations (BIE on the surface and PDE in the volume). This is mainly due to the fact that, with the time harmonic hypothesis, the mesh size is constrained by the wavelength of the illuminating wave. Typically, for linear finite elements, ten discretization points per wavelength are required to avoid "pollution effects" [3] and obtain an accurate enough solution [4]. Consequently the size of the discretized problems grows as a cubic (square) function of the frequency in the volume (resp. on the surface where the BIE is defined); for large objects or high frequency illuminating waves the size of the associated linear systems can quickly increase up to billions of unknowns.

Beyond the size of the linear systems, the last but not least difficulty is related to their numerical properties. The linear systems exhibit a natural 2×2 block structure induced by the discretization in the 3D volume and the discretization on the surface (BIE). The diagonal block associated with BIE is dense, the other one is sparse. This latter block does not have any structure (reflecting the unstructured mesh) and does not enjoy any suitable numerical property. This matrix is complex symmetric (non-Hermitian) indefinite with a very large "near null space" [17]. Furthermore, it is badly scaled due to the high contrast between dielectric characteristics of coating materials impacting the already poor condition number. In this paper we focus on the solution of the diagonal block associated with the discretization in the volume.

Among the most promising numerical techniques for the scalable solution of PDE are the multigrid approaches [13]. For elliptic problems, their robustness and effectiveness have been clearly demonstrated as well as their good parallel efficiency on a large number of cores [2]. For this type of PDE similar approaches where the unstructured mesh is automatically refined [26] within a multigrid solver is reported in [6, 21] where parallel performances on huge problems are presented. However, obtaining similar performance on indefinite problems, such as Helmholtz and Maxwell, is still an active research area. In the last decade many attempts have been made. In order to overcome the weakness of the usual relaxation techniques used as smoother, the use of GMRES [40] has been proposed either on the coarser levels as soon as the indefiniteness grows [14] or straight from the finest grid [19]; alternative smoothers based on Chebyshev polynomial [1], ILU(0) [38] or conjugate gradient applied to the normal equations [29] have also been considered. In [15, 19, 29], the authors propose to tackle the problem by using special construction and solution techniques for the coarse problem. Other techniques based on either different physical decompositions (including wave ray equation [32] or product of plane waves with advection-diffusion-reaction solution [22]) or on mathematical transformation (like Helmholtz decomposition that conducts to a separate treatment of the kernel of the *curl* operator

and its complement [20, 25]) have been investigated. Alternatively, the introduction of auxiliary unknowns to obtain a first order system is considered in [31]. Other authors use multigrid as preconditioner for Krylov subspace methods. Following the pioneer work of [35], shifted Laplacian techniques are proposed in [16, 41]. Many studies are conducted in the framework of algebraic multigrid [7, 12, 27, 28] in real arithmetic that have no straightforward extension to complex calculation for defining an efficient aggregation procedure [9, 34, 37, 33].

In our work, we intend to further exploit one feature of the problem we intend to solve. That is, the geometry of the target and coating is sufficiently well described by a mesh of moderate size, where the mesh size does not comply with the ten points per wavelength condition. This coarse mesh can play the role of the coarsest grid for a full geometric multigrid. The other meshes of the hierarchy can be automatically generated by successively refining each element until the wavelength rule is satisfied on the finest mesh. Such an approach has another clear advantage when one is interested in a scanning frequency study, that is when the RCS are computed for a given object varying the frequency of the illuminating waves. The same coarse mesh, which construction is a heavy human time consuming task, can be reused for a large enough range of the frequencies of interest.

The solution of the Maxwell equations in the frequency domain is not the only framework where a mesh, that approximates sufficiently the geometry, is not fine enough to comply with numerical and physical constraints associated with the PDE to be solved. Still in the wave propagation context, this situation can be encountered in seismic-imaging for oil reservoir research where acoustics waves upon a wide frequency band [39] need to be simulated. Other examples exist in fluid applications with viscous layers in laminar Navier Stokes or successive Kolmogorov cascades of vortices in turbulent flows solved by Direct Numerical Simulations. More generally, all numerical simulations that implement Adaptive Mesh Refinement techniques are other examples where the mesh refinement is not governed by geometry constraints.

The paper is organized as follows. In Section 2 we give some background on the problem to be solved and the numerical techniques involved. The numerical and algorithmic formulation of all the components of the parallel full multigrid solver is described in Section 3. In Section 4, we investigate the numerical features of the full multigrid scheme on problems exhibiting different difficulties. In particular, we are interested in increasing the wave number when the material electromagnetic index or (and) the frequency grows. Section 5 is devoted to the assessment of the parallel scalability of our implementation where we report performance on a problem with up to 1.3 billion unknowns solved using only one thousand cores.

2 Some background on time harmonic Maxwell's equations

With the time harmonic assumption, all the fields have a periodic behavior in time. The RCS calculation requires the solution of the following external problem

$$\begin{cases} \nabla \wedge \mathbf{E} - i\omega\mu\mathbf{H} = 0 & \text{in } \mathbf{T}_{ext}, \\ \nabla \wedge \mathbf{H} + i\omega\epsilon\mathbf{E} = 0 & \text{in } \mathbf{T}_{ext}, \\ (\mathbf{E} \wedge \nu + (\mathbf{H} \wedge \nu) \wedge \nu) = R(-\mathbf{E} \wedge \nu + (\mathbf{H} \wedge \nu) \wedge \nu) + \mathbf{g} & \text{on } \Gamma_{int}, \\ (-\mathbf{E} \wedge \mathbf{e}_r + (\mathbf{H} \wedge \mathbf{e}_r) \wedge \mathbf{e}_r) = \mathcal{O}\left(\frac{1}{r^2}\right) & \text{when } r \text{ is growing to infinity,} \end{cases} \quad (1)$$

where \mathbf{E} and \mathbf{H} are the electric and magnetic fields, ω the pulsation, ϵ and μ the permittivity and permeability of the local material, g is a current source term and \mathbf{e}_r the outer normal vector to the sphere enabling to express the radiation condition at infinity. Γ_{int} is a closed internal surface inside the target where the reflection operator R is known (for instance a perfectly conducting surface where $R = 1$), \mathbf{T}_{ext} is the complete space surrounding Γ_{int} , ν is the outgoing normal to

the frontier Γ_{int} (see Figure 1). The last equation in (1) stands for the radiation condition at infinity.

Introducing relative permittivity and permeability $\epsilon = \epsilon_r \epsilon_0$ and $\mu = \mu_r \mu_0$, the problem can be written in its second order PDE form

$$\begin{cases} \nabla \wedge \left(\frac{1}{\mu_r} \nabla \wedge \mathbf{E} \right) - i \mathbf{k}_0^2 \epsilon_r \mathbf{E} = 0 & \text{in } \mathbf{T}_{ext}, \\ \mathbf{H} = -\frac{1}{i\omega} \mu_r \nabla \wedge \mathbf{E}, \end{cases} \quad (2)$$

where $\mathbf{k}_0 = \omega \sqrt{\epsilon_0 \mu_0}$ is the wave number in the vacuum. We introduce also the local wave number in each material that plays an important role in the rest of the paper

$$\mathbf{k} = \mathbf{k}_0 \sqrt{\epsilon_r \mu_r} = \mathbf{k}_0 \mathbf{n} = 2\pi f \mathbf{n} \sqrt{\epsilon_0 \mu_0}, \quad (3)$$

\mathbf{n} being the electromagnetic index, f the frequency of the illuminating incident wave. ϵ_r , μ_r may include imaginary parts that are referred to as the "electromagnetic losses".

To take into account the radiation condition at infinity without any approximation a BIE has to be retained. This BIE is defined on a surface Γ_{ext} that can be the external surface of the target itself or an external one in the space near around the target. That BIE method leads to the calculation of the Electric and Magnetic currents \mathbf{J} , \mathbf{M} on the frontier Γ_{ext}

$$\begin{cases} \mathbf{J} = -\mathbf{H} \wedge \boldsymbol{\nu}, \\ \mathbf{M} = \mathbf{E} \wedge \boldsymbol{\nu}. \end{cases} \quad (4)$$

Practical details on BIE and its implementation can be found in [36].

2.1 A strong coupling

The solution of Equation (2) requires an internal solution in the domain Ω that lies between Γ_{int} and Γ_{ext} as well as an external one for the BIE on Γ_{ext} . The internal problem is solved by a PDE method with a finite element discretization (tetrahedra). Its variational formulation reads

$$\int_{\Omega} [(1/\mu_r) \text{curl} \mathbf{E} \text{ curl} \mathbf{E}^* - \mathbf{k}_0^2 \epsilon_r \mathbf{E} \mathbf{E}^*] d\Omega + \int_{\Gamma_{ext}} \mathbf{H} \wedge \mathbf{E}^* \boldsymbol{\nu} d\gamma = 0 \quad (5)$$

where \mathbf{E}^* denote the test functions. Introducing Equations (4) into the second term of above equation induces a strong coupling between internal degrees of freedom (dof) \mathbf{E} in the volume and the ones, \mathbf{J} and \mathbf{M} , on the surface. The matrix arising from the discretization has the following form 2×2 block structure:

$$\left(\begin{array}{c|cc} \mathbf{A}_{11}^v & \mathbf{A}_{12}^v & 0 \\ \hline \mathbf{A}_{21}^v & (\mathbf{A}_{22}^v + \mathbf{A}_{22}^s) & \mathbf{A}_{23}^s \\ \hline 0 & \mathbf{A}_{32}^s & \mathbf{A}_{33}^s \end{array} \right) \begin{pmatrix} \mathbf{E}_{int} \\ \mathbf{M} \\ \mathbf{J} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \end{pmatrix}. \quad (6)$$

The (1,1) block \mathbf{A}_{11}^v results from the finite element discretization in the interior volume Ω and is sparse. The (2,2) block, associated with \mathbf{J} and \mathbf{M} on the surface Γ_{ext} is dense with the contribution of matrices denoted with a superscript "s" coming from the BIE discretization. The matrices \mathbf{A}_{12}^v , \mathbf{A}_{21}^v , \mathbf{A}_{22}^v account for the strong coupling between \mathbf{E} , \mathbf{J} and \mathbf{M} . Because the (1,1) block is associated with unknowns in the volume, its size is usually significantly larger than the (2,2) block associated with unknowns defined on the surrounding surface.

In order to solve Equation (6) a Schur complement approach might be considered that consists in eliminating the electric field \mathbf{E}

$$[\mathbf{A}_{22}^v - \mathbf{A}_{21}^v (\mathbf{A}_{11}^v)^{-1} \mathbf{A}_{12}^v + \mathbf{A}_{22}^s] \mathbf{M} + \mathbf{A}_{23}^s \mathbf{J} = \mathbf{b}_2 - \mathbf{A}_{21}^v (\mathbf{A}_{11}^v)^{-1} \mathbf{b}_1$$

and

$$\mathbf{A}_{32}^s \mathbf{M} + \mathbf{A}_{33}^s \mathbf{J} = \mathbf{b}_3.$$

One of the main drawback of this approach is the huge computational resource consumption required by the calculation of $(\mathbf{A}_{11}^v)^{-1} \mathbf{A}_{12}^v$. Even though state of the art parallel sparse direct solver such as PaStiX [24] can compute this Schur complement using compute intensive kernels, the computing time is prohibitive for the huge problems targeted in our studies.

2.2 A weak coupling

At the continuous level an alternative solution technique consists in using a domain decomposition methodology well suited to this problem [?]. The domain Ω is partitioned in a set of embedded subdomains Ω_i as displayed in Figure 1. The inner most subdomains are numbered first. Along

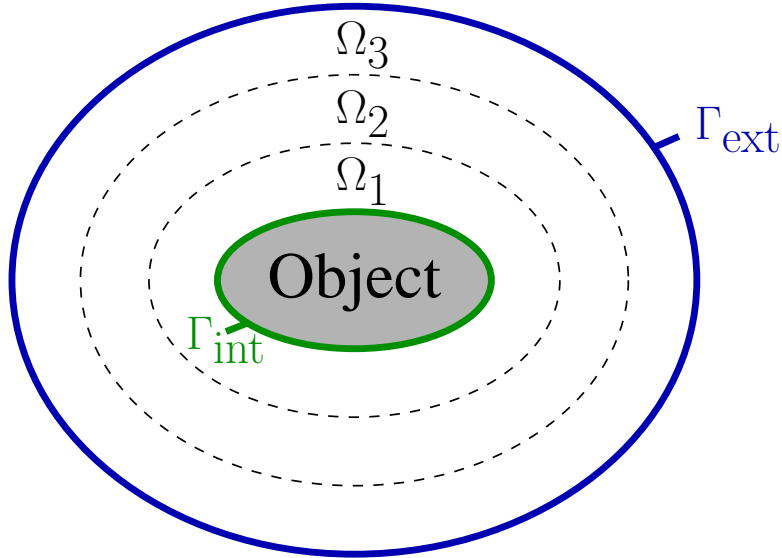


Figure 1: Bulb peel domain decomposition.

the interfaces between the subdomains a transmission condition is used to bind the electric field on the both sides of the interface. This condition is imposed on all the interfaces including the external surface Γ_{ext} that is viewed as a regular subdomain. This setting enables us to define a multiplicative Schwarz iterative process where we sequentially sweep over the subdomains as:

$$\mathbf{E}_{\Omega_i}^{k+1} = f(\mathbf{E}_{\Omega_{i-1}}^{k+1}, \mathbf{E}_{\Omega_i}^k, \mathbf{E}_{\Omega_{i+1}}^k). \quad (7)$$

For each subdomain, the boundary condition for the inner (outer) interface is given by the transmission condition defined with its internal (resp. external) neighbor at iteration $k+1$ (resp. k). For all the subdomains, but Γ_{ext} , the PDE (2) is discretized with boundary conditions defined by the transmission condition associated with the domain decomposition scheme.

In a preliminary work, the large sparse linear systems in each subdomain Ω_i were solved using a parallel sparse direct solver [24]; the dense system associated with the BIE on Γ_{ext} was solved iteratively using a multilevel fast multipole method to perform the matrix-vector product [36].

2.3 The design process for complex objects

In an industrial context, the Maxwell solver presented in the previous sections is used as the main computational kernel in a set of nested loops. When a new object is designed, its study consists in computing its RCS for a set of frequencies and for each of them for various angles of the illuminating wave. The global algorithm reads as depicted in Algorithm 1. As it can be seen in Equation (5), an important point for the following is that the right-hand side depends on the frequency and the illumination angles while the matrix coefficients depend only on the frequency. The sparsity pattern of the matrix is induced by the mesh connectivity (initial coarse mesh in this case).

Algorithm 1 The loops in the design process

```

1: % Scan the frequency interval of interest
2: for  $f = freq_{min}, freq_{max}$  do
3:   % Scan the angles of interest for the illuminating wave
4:   for  $\theta = \theta_{min}, \theta_{max}$  do
5:     Solve  $A(f) x(f, \theta) = b(f, \theta)$ 
6:   end for
7: end for

```

With the state of the art PaStiX [24] direct solver used, the largest problem solved in each subdomain was 82 million unknowns on a platform by using 48 SMP nodes each of them with 16 cores and a large amount of memory per node. The current tendency on emerging parallel platforms is to shrink the amount of memory per core which might lead to reduce the size of the subdomains. For a given problem size, it implies increasing the number of subdomains which numerically induces an increase of the number of iterations of the multiplicative Schwarz scheme defined by Equation (7).

Consequently designing a new and more scalable parallel sparse linear solver is necessary to enhance the robustness and efficiency of the domain decomposition solver and to reach problem sizes up to several billion unknowns.

3 The full geometric multigrid solver

The calculation in a full multigrid solver might be decomposed into two main phases. In the first phase, starting from a linear problem defined on the coarsest grid, it generates an initial guess on the finest grid. The second phase consists in performing classical multigrid cycles to refine the approximate solution. The full geometric multigrid we consider in this work is summarized in Algorithm 2, where $P_{\ell-1}^\ell$ denotes the prolongation operator from the mesh $G_{\ell-1}$ to the finer mesh G_ℓ and $S_\ell^\nu(A_\ell, x_\ell^0, b_\ell)$ denotes ν steps of the smoother performed on G_ℓ starting from the initial guess x_ℓ^0 with the right-hand side b_ℓ and the matrix A_ℓ constructed by discretizing the Maxwell equations on G_ℓ . We emphasize that in our implementation no multigrid cycle is performed when we move up the initial guess as performing V-cycle in moving up the grid hierarchy did not exhibit any numerical benefit but a small computing penalty. At each level the interpolated approximation of the solution is only improved via ν steps of the smoother.

The regular $MG(A_{\ell_{max}}, x_{\ell_{max}}^{iter}, b_{\ell_{max}}, P_{\ell-1}^\ell, R_{\ell-1}^{\ell-1}, \nu_1, \nu_2)$ performs a cycle of geometric multigrid, where $R_{\ell-1}^{\ell-1}$ is the restriction operator from mesh G_ℓ down to grid $G_{\ell-1}$, ν_1 (ν_2) denotes the number of pre-smoothing (respectively post-smoothing) steps. On each mesh level, the Maxwell equations are discretized to compute A_ℓ . The solution on the coarsest mesh is performed by a parallel sparse direct solver, which data distribution drives the mesh partition and parallel calculation on the hierarchy of finer meshes (the parallel implementation is described in more details in Section 5). To complete the numerical description of our scheme, three central computational kernels still need to be defined that are the grid transfer operators ($R_{\ell-1}^{\ell-1}, P_{\ell-1}^\ell$) and the smoother $S_\ell^\nu(A_\ell, x_\ell^0, b_\ell)$.

Algorithm 2 Full multigrid algorithm $[A_0, b_0]$: in, $[x_{\ell_{max}}]$: out

```

1: % Generate the initial guess on the coarsest mesh
2: Solve  $A_0 x_0^1 = b_0$  on the coarsest mesh by using a sparse direct solver, where  $b_0 = R_1^0(\dots(R_{\ell_{max}}^{\ell_{max}-1}(b_{\ell_{max}})))$ 
3: % Move up the approximation from the coarsest to the finest mesh
4: for  $\ell = 1, \ell_{max}$  do
5:   Compute  $A_\ell$  that discretizes the PDE on  $G_\ell$ 
6:   Interpolate the current approximation from  $G_{\ell-1}$  on  $G_\ell$  :  $x_\ell^0 = P_{\ell-1}^\ell(x_{\ell-1}^1)$ 
7:   Improve the approximation on current mesh  $G_\ell$  via  $\nu_0$  smoothing iterations :  $x_\ell^1 = S_\ell^\nu(A_\ell, x_\ell^0, b_\ell)$ 
8: end for
9: % Perform regular multigrid cycle from the initial guess  $x_{\ell_{max}}^1$ 
10: while  $iter = 1$  until  $\|A_{\ell_{max}} x_{\ell_{max}}^{iter} - b_{\ell_{max}}\| / \|b_{\ell_{max}}\|$  small enough do
11:   % Perform a regular multigrid cycle
12:    $x_{\ell_{max}}^{(iter+1)} = MG(A_{\ell_{max}}, x_{\ell_{max}}^{iter}, b_{\ell_{max}}, P_{\ell-1}^\ell, R_{\ell-1}^{\ell-1}, \nu_1, \nu_2)$ 
13: end while

```

3.1 Grid transfer operators

3.1.1 The prolongation operator $P_{\ell-1}^\ell$

The prolongation operator can be defined element-wise by considering the Nédélec first order finite element definition. The test functions used to build the basis on two successive meshes are depicted in Figure 2. Each degree of freedom α_i associated with the edge Γ_i with unit vector $\vec{\tau}_i$ is defined by the equation

$$\alpha_i = \int_{\Gamma_i} \vec{E}(x, y, z) \cdot \vec{\tau}_i \, d\Gamma_i.$$

The electric field $\vec{E}(x, y, z)$ at any point within the element can be expressed in the basis as

$$\vec{E}(x, y, z) = \sum_i \vec{p}_i(x, y, z) \alpha_i$$

where \vec{p}_i denotes the Nédélec first order basis function associated with edge Γ_i defined such that for each edge Γ_j

$$\int_{\Gamma_i} \vec{p}_j \cdot \vec{\tau}_i \, d\Gamma_i = \delta_{i,j}.$$

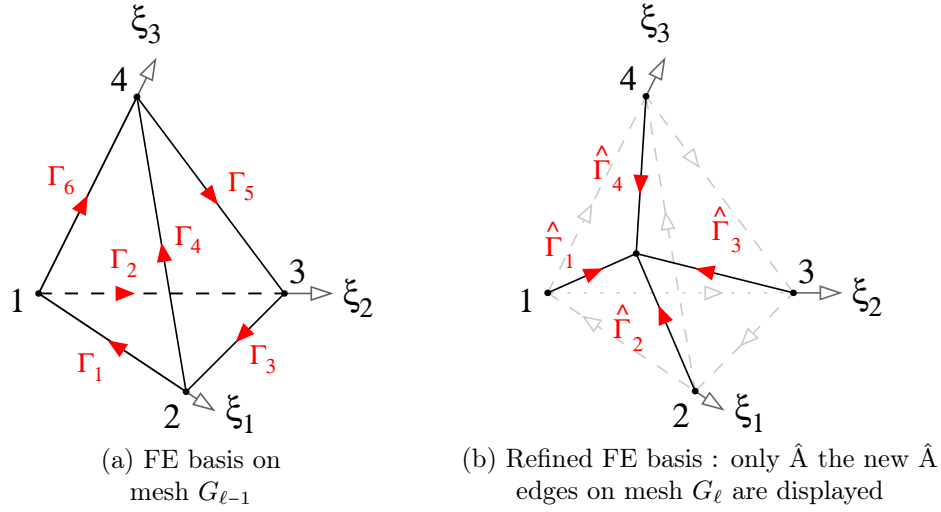


Figure 2: 3D Nédélec edge finite element.

According to the orientation convention of the basis element, each first order basis function \bar{p}_i reads

$$\begin{aligned} \bar{p}_1 &= \begin{pmatrix} -1 + \xi_2 + \xi_3; & -\xi_1; & -\xi_1 \end{pmatrix}, \\ \bar{p}_2 &= \begin{pmatrix} \xi_2; & 1 - \xi_1 - \xi_3; & \xi_2 \end{pmatrix}, \\ \bar{p}_3 &= \begin{pmatrix} \xi_2; & -\xi_1; & 0 \end{pmatrix}, \\ \bar{p}_4 &= \begin{pmatrix} -\xi_3; & 0; & \xi_1 \end{pmatrix}, \\ \bar{p}_5 &= \begin{pmatrix} 0; & \xi_3; & -\xi_2 \end{pmatrix}, \\ \bar{p}_6 &= \begin{pmatrix} \xi_3; & \xi_3; & 1 - \xi_1 - \xi_2 \end{pmatrix}, \end{aligned}$$

where the ξ_i denotes the coordinates within the tetrahedron. Based on these definitions and considering the edges between each node and the center of gravity $C = \frac{1}{4}(1, 1, 1)$, the prolongation operator can be computed. Considering the new edge $\hat{\Gamma}_1$ between nodes 1 and C with unit vector $\hat{\tau}_1 = \frac{1}{\sqrt{3}}(1, 1, 1)$, the new degree of freedom γ_1 associated with the edge $\hat{\Gamma}_1$ is defined by the following relation:

$$\gamma_1 = \int_{\hat{\Gamma}_1} \vec{E}(x, y, z) \cdot \hat{\tau}_1 d\hat{\Gamma}_1 = \sum \alpha_i \left[\int \bar{p}_i(x, y, z) \cdot \hat{\tau}_1 d\hat{\Gamma}_1 \right].$$

Using the parametric definition of $\hat{\Gamma}_1(t) = \frac{t}{4}(1, 1, 1)$ with $d\hat{\Gamma}_1 = \frac{\sqrt{3}}{4} dt$ and integrating each component, we obtain

$$\gamma_1 = \frac{1}{4} \sum \alpha_i \left[\int_0^1 \bar{p}_i\left(\frac{t}{4}, \frac{t}{4}, \frac{t}{4}\right) \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} dt \right] = \frac{1}{4}(-\alpha_1 + \alpha_2 + \alpha_6).$$

Using the same technique for each new degree of freedom enables us to define the degree of freedom on the new edges of G_{ℓ} by

$$\begin{aligned} \gamma_1 &= \frac{1}{4} \begin{pmatrix} - & \alpha_1 & + & \alpha_2 & + & \alpha_6 \end{pmatrix}, \\ \gamma_2 &= \frac{1}{4} \begin{pmatrix} + & \alpha_1 & - & \alpha_3 & + & \alpha_4 \end{pmatrix}, \\ \gamma_3 &= \frac{1}{4} \begin{pmatrix} - & \alpha_2 & + & \alpha_3 & - & \alpha_5 \end{pmatrix}, \\ \gamma_4 &= \frac{1}{4} \begin{pmatrix} - & \alpha_4 & + & \alpha_5 & - & \alpha_6 \end{pmatrix}. \end{aligned}$$

The prolongation is the identity for the edges that belong to both G_{ℓ} and $G_{\ell-1}$.

3.1.2 The restriction operator $R_\ell^{\ell-1}$

In this work we consider the canonical injection to define the restriction operator. In order to preserve the symmetry of the multigrid operator, it would have been preferable to consider $R_\ell^{\ell-1} = (P_{\ell-1}^\ell)^T$ so that the value on the coarse edge accounts for the values on the surrounding fine edges. Although simple to consider in a mathematical setting, its parallel implementation is slightly more complex and costly; consequently we did not consider it in this work. The numerical penalty is the symmetry lost, which prevents us to use symmetric Krylov subspace solvers [18] when multigrid is used as preconditioner for some numerical experiments.

3.2 Matrix-free smoothers

The objective of this work is to design an efficient parallel solver able to solve very large 3D problems. In order to reduce as much as possible the memory footprint we only consider parallelizable smoothers that can be implemented in a matrix-free fashion and with very small memory requirements. In our finite element framework, the matrix-vector product can be computed without forming explicitly the complete matrix. The matrix-vector product is rather computed by summing the elementary matrix-vector products. Possible candidate smoothers are matrix polynomial [1] or unsymmetric Krylov solvers with short term recurrences. After an evaluation of all identified candidates on the operator considered here, we did not notice any relevant benefit over weighted Jacobi. Consequently we have chosen this last technique that requires only the extra storage of the diagonal of the complete matrix.

The recently proposed L1-Jacobi smoother [2] does not neither require a larger storage. It appears to be well suited for definite and semi-definite Maxwell problems [30]. We have investigated its efficiency on the indefinite Maxwell problem studied in this paper. When the wave number is increased, FMG using L1-Jacobi as smoother converges slightly better than FMG using weighted (damped) Jacobi as the problem is just departing from positive definiteness. However this advantage vanishes very quickly when the indefiniteness increased up to the point where the two smoothers do not longer ensure the convergence of FMG. For this reason we have chosen to keep the weighted Jacobi scheme as the smoother in FMG used as a fixed point solver or in MG used as a preconditioner of a Krylov solver.

4 Numerical experiments

In this section we investigate the numerical behaviour of the FMG solver described in the previous sections. We first study in Section 4.1, the effect of the damping parameter of the Jacobi smoother on the convergence. Then in Section 4.2 we investigate how the physical parameters, namely the electromagnetic index and the permittivity and permeability of the local material, affect the convergence of the scheme. Even though large electromagnetic indices prevent the convergence of FMG we show that the associated MG scheme can still be an extremely effective preconditioner for these electromagnetic problems.

In order to assess the correctness of the computed solutions we consider a set of test problems where the analytical solution of the associated RCS is known via the Mie's series theory. These still realistic problems consist of a coated sphere where both the electromagnetic characteristics of the coating material and the frequency of the illuminating incident wave are varied. With respect to the notation depicted in Figure 1, the perfectly electric conducting surface of the sphere defines the internal interface Γ_{int} , the coating material defines Ω_1 that is surrounded by a layer of vacuum Ω_2 . This latter layer enables us to move slightly the BIE from the surface of the

material. The numerical behaviour of FMG is investigated for the solution of the electromagnetic problem in Ω_1 where a coarse mesh with 14,565 dof is provided by a mesh generator.

In Figures 3 and 4 we display the convergence history of the residual norm scaled by the norm of the right-hand side and adopt the following notations:

- crosses (\times) represent the initial scaled residual before each V-cycle,
- empty squares (\square) represent the scaled residual after each iteration of the smoother,
- solid squares (\blacksquare) represent the residual after the coarse grid correction, that is just after the error has been prolonged on the finest mesh and before the final ν_2 smoothing iterations of the cycle are performed.

Because we display the convergence history for experiments with $\nu_1 = \nu_2 = 2$, each V-cycle on the curves is composed by the following sequence of symbols: " $\times \square \square \blacksquare \square \square$ ". These different computational kernels are referred to as multigrid steps in the Figures.

4.1 On the relaxation parameter of the Jacobi smoother

The numerical experiments reported in this section correspond to two-level FMG scheme. The solved electromagnetic problem has 57,909 dof and an electromagnetic index equal to 1.41. In Figure 3 we display the convergence history when the damping parameter of the Jacobi smoother is varied. As usual for the Maxwell and Helmholtz indefinite problems, only relaxation parameters lower than one have been considered. It can be seen that the speed of convergence of FMG is strongly influenced by the damping parameter ω . In that example, with a damping parameter higher than 0.6 FMG diverges and the best convergence is achieved with $\omega = 0.4$.

4.2 Influence of the wave number

In order to study how the convergence is affected by the electromagnetic properties of the simulation, we investigate in this section the convergence history of FMG when the electromagnetic index varies for material with and without losses. The numerical experiments reported in this section correspond to three-level FMG. The electromagnetic problem that is solved has 231,045 dof. We display in Table 1, the permittivity (ϵ) and permeability (μ) as well as the corresponding electromagnetic index (n) and wave number (k). The first three rows correspond to lossless materials that do not exhibit neither dielectric nor magnetic losses, while the last three correspond to material with dielectric and magnetic losses. The first three columns give the number of mesh points per wavelength λ for each of the three level of meshes as well as the number of dof on each level (in parenthesis we give the ratio of the average edge length of the mesh with respect to the same quantity on the coarsest mesh). Level 0 corresponds to the coarsest mesh provided by a mesh generator, level 1 (level 2) is the mesh generated by the FMG solver using one refinement step (resp. two refinement steps).

In Figure 4 we display the convergence history of the first ten iterations of FMG on the linear systems associated with the material with losses. It can be seen that for the three electromagnetic indices, FMG does converge. The attainable accuracy in term of scaled residual norm depends on the electromagnetic indices. Nevertheless, we have checked that the final RCS computed from the solution of the linear systems perfectly match the ones given by the analytical solutions [10]. It is important to notice that for n equal to 5.64, the coarsest mesh is far from complying with the ten points per wavelength rule, while the solution on the finest mesh (that only weakly complies with this rule) is physically correct (v.s. its RCS value). Similar results are reported in Figure 5

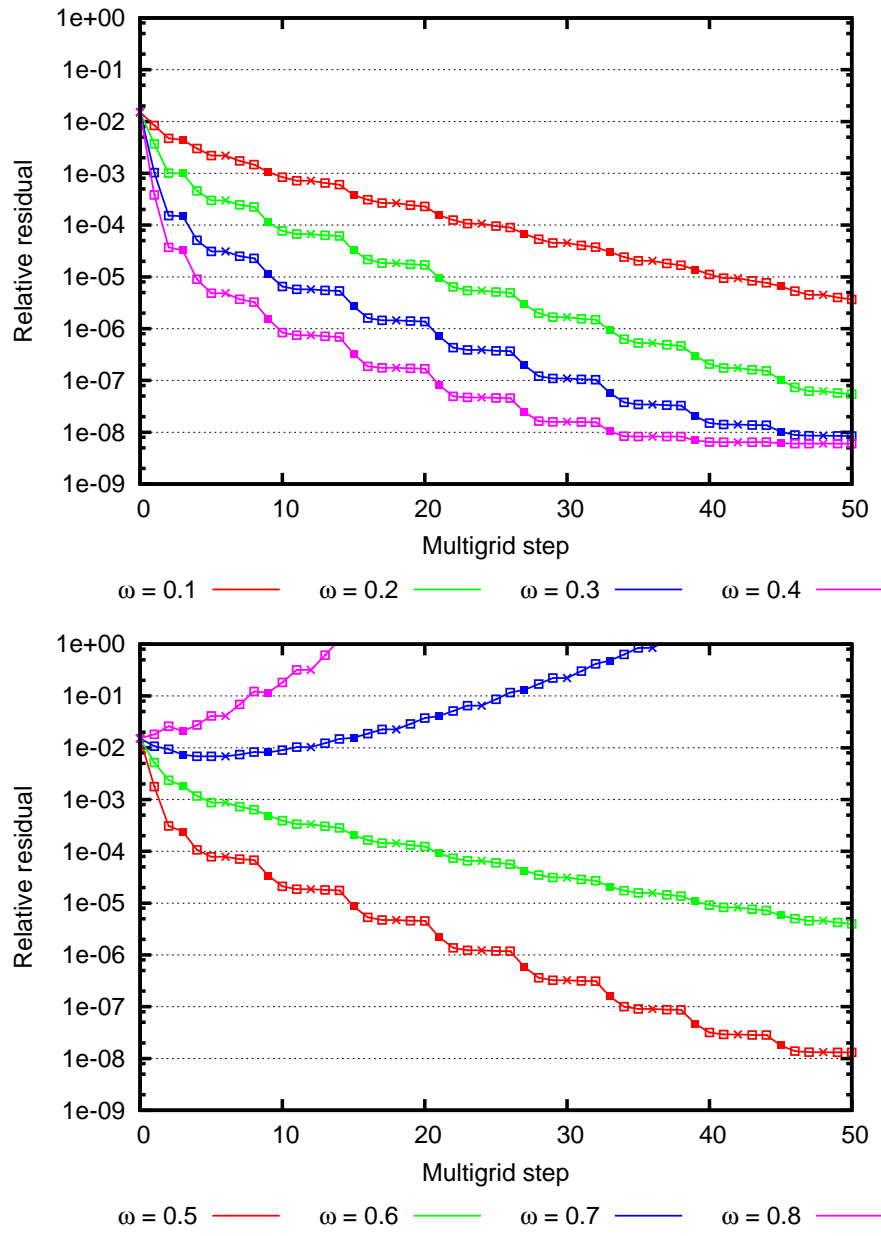
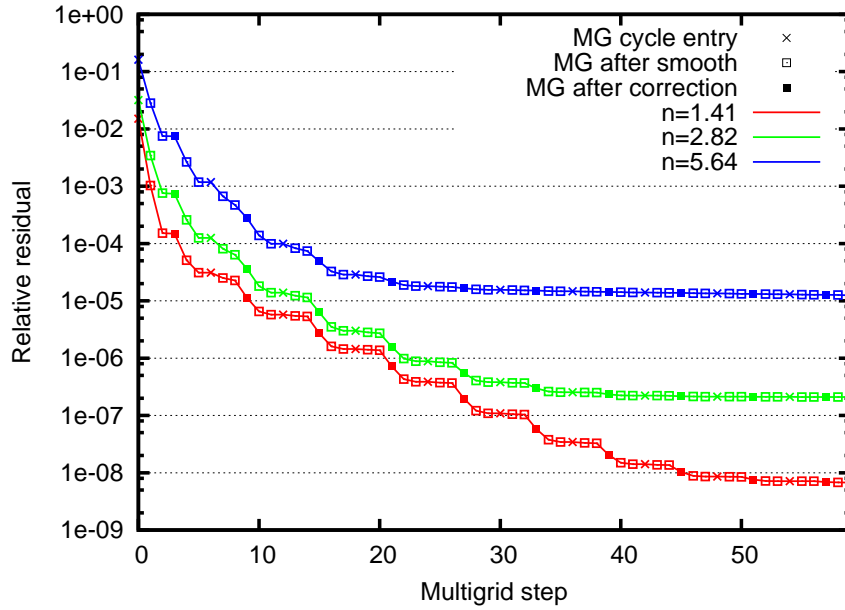


Figure 3: Convergence history when the Jacobi damping parameter is varied (V-cycle with $\nu_1 = \nu_2 = 2$).

# points per λ			electromagnetic parameters			
level 0	level 1	level 2	ϵ_r	μ_r	$ n $	$ k $
12	17	23	2	1	1.41	4.44
6	9	11	4	2	2.82	8.89
3	4	6	8	4	5.64	17.77
12	17	23	1+1i	1+1i	1.41	4.44
6	9	11	2+2i	2+2i	2.82	8.89
3	4	6	4+4i	4+4i	5.64	17.77
# dof						
14,625 (1)	57,909 (0.73)	231,045 (0.54)				

Table 1: Characteristics of the test problem.

Figure 4: Convergence history for material with losses for various electromagnetic indices (10 FMG V-cycles with three levels and $\nu_1 = \nu_2 = 2$).

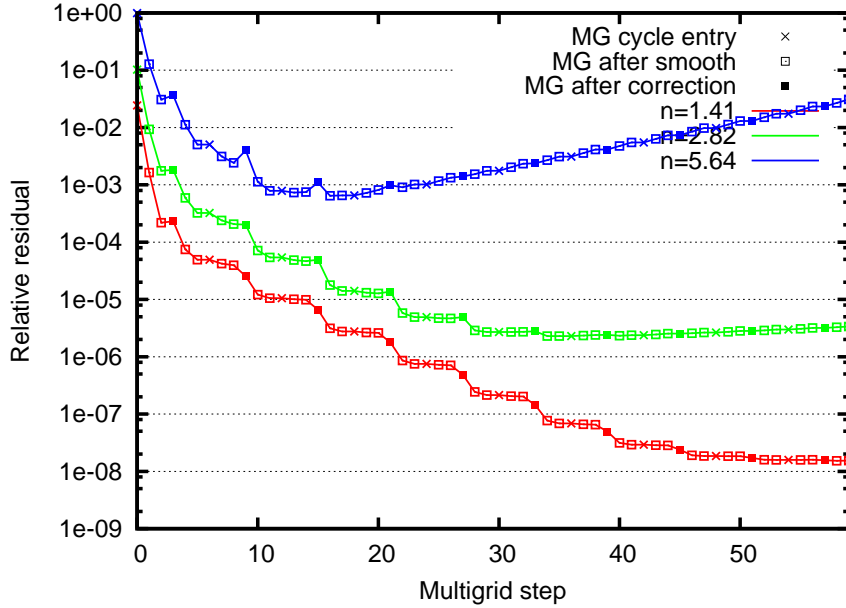


Figure 5: Convergence history for lossless material for various electromagnetic indices (10 FMG V-cycles with three levels and $\nu_1 = \nu_2 = 2$).

for the same electromagnetic indices but for materials without loss. It can be seen that except for the smallest index, FMG does not converge.

Similar observations are studied for the solution of the Helmholtz equations [14, 15] in a simpler setting (regular discretization on the unit square geometry and simpler boundary conditions). The main reason is that the indefiniteness of the matrices increases when the wave number is increased; consequently the iterative solution of the associated linear systems become more challenging. Our results extend these observations to the Maxwell equations on more complex geometries. The better numerical behaviour for material with losses has also been observed for the Helmholtz equations [15] where it is argued that losses induce a shift of the matrix spectrum in the imaginary plane making easier the iterative solution of the associated linear systems.

4.3 MG as preconditioner

As it can be seen in Figure 5 even when FMG eventually diverges, it always exhibits a residual norm decrease in the first iterations. This observation motivates the use of MG as a preconditioner for a Krylov subspace method [14, 16]. We display in Figure 6 and 7 the convergence history of GMRES [40] for the solution of the most challenging problems considered in the previous section. Namely, we select the largest problems with the highest electromagnetic index and consider material with and without loss. In both cases, we used one V-Cycle of MG as right preconditioner. It can be seen that preconditioned GMRES converges with a small restart even for the solution of the problems where FMG did not converge. GMRES without preconditioner either stagnates or converges very slowly. As it could have been expected, the convergence is faster for the problem with losses (Figure 6) than for the problem without loss (Figure 7). Although not shown in this paper, we also performed experiments using more than one V-cycle in the preconditioner; it did not improve significantly the convergence speed while it increased

slightly the computational time.

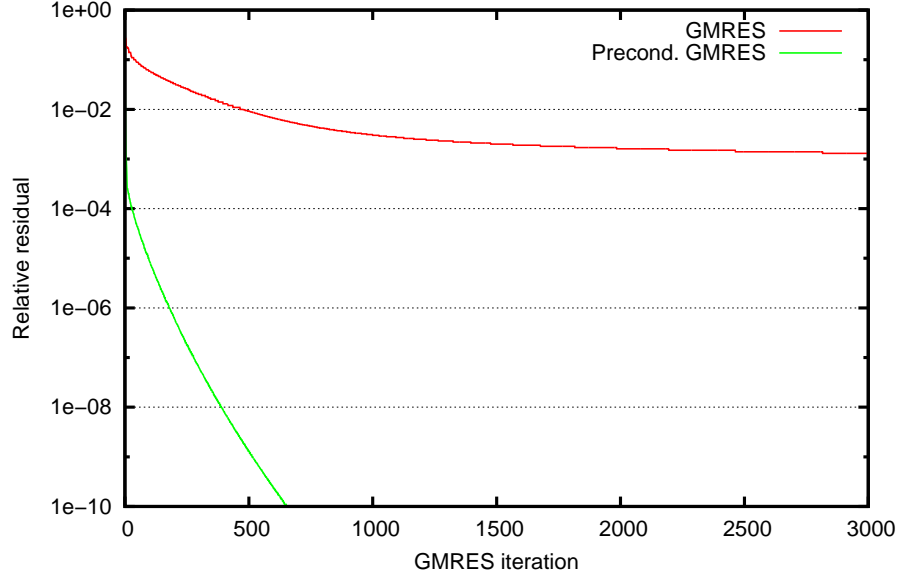


Figure 6: Convergence history of unpreconditioned and FMG-preconditioned GMRES(10) in the case of material with losses.

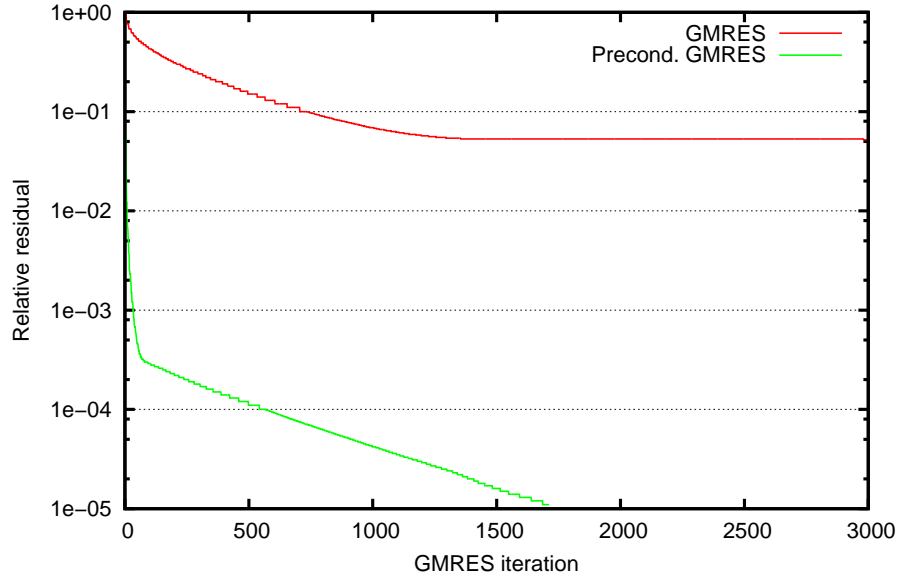


Figure 7: Convergence history of unpreconditioned and FMG-preconditioned GMRES(10) in the case of lossless material.

5 Parallel implementation and performance

In this section, we first present in Section 5.1 the governing ideas that have driven the parallel implementation of FMG scheme. In particular we describe the selected data distribution for the implementation of the main computational kernels that are the smoother, the restriction/prolongation and the coarse grid solution. Then in Section 5.2, we report on the parallel performances of this implementation.

5.1 Parallel Implementation

When parallelizing a PDE solver that relies on a finite element technique a natural approach consists in distributing the set of elements among the computing nodes. Using a partitioning of the coarse mesh, our refinement strategy can be applied to generate the hierarchy of meshes without any communication. This element partitioning implies that the dof along the interfaces generated by the partitioning are replicated on different processors. In our case, first order Nédélec edge finite elements are used so that the unknowns associated with the edges along the interfaces are duplicated.

Parallel matrix free smoother With this partitioning of the tetrahedron elements, it is easy to perform a parallel matrix-vector using an element-by-element calculation without explicitly forming the local matrices associated with each partition subset (only the elementary ones). Once this parallel computation has been performed, the result of the matrix-vector product is known for all the interior entries; then a few neighbour to neighbour exchanges along the interfaces enable the final assembly of the matrix-vector results for the entries on these interfaces. Notice that for our refinement policy, the volume of communication (and the edges involved in the communication) remains identical at each level of the mesh hierarchy. Using this parallel scheme, we can easily implement a matrix-free calculation of the damped Jacobi smoother that reads

$$x^{k+1} = x^k + \omega D^{-1} \left(b - \sum_{e \in \text{set of element}} A^e x^k \right)$$

where A^e denotes the elementary matrices and the exchanges along the interfaces are embedded in the sum. Finally, it is worth noticing that the assembly of the diagonal entries of the matrix D can be computed using a very similar scheme as the one used for the matrix-vector product requiring the same communication pattern.

Grid transfer operators With the data distribution used by the parallel smoother implementation the grid transfer operators are straightforward to compute. With our mesh refinement strategy the prolongation is local to the elements and consequently its calculation is embarrassingly parallel and does not require any communication. We consider a canonical restriction in our scheme. Consequently, because the unknowns along the interfaces are replicated, the restriction can be computed without any communication at the slight expense of the redundant calculation of the restriction for each edge on the interface on all computing processors sharing these edges.

Coarse grid solution Considering our approach to solve Maxwell problems, the multigrid scheme requires a very robust solution on the coarse mesh. To achieve this objective with a sufficient parallel efficiency, we have chosen the parallel sparse direct solver PaStiX for this step; this solver has been proved to be reliable in the Maxwell problem scope. In that framework the memory peak and the most computationally complex numerical kernel is the matrix factorization on the coarse mesh and the forward and backward substitutions using the factors. The parallel

implementation of the direct solver is the most critical part of our multigrid method and we decided that the data distribution for the above mentioned other kernels will be entirely deduced from the data distribution for the sparse parallel direct solver. Notice that the parallel implementations of the other kernels are based on a distribution of the elements, while the matrix entries (as well as the solution and right-hand side entries) are associated with the edges. We then need to implement an effective strategy enabling us to easily move from an edge distribution for the direct solver to an element distribution for all the other kernels.

Strategy to move from an edge to an element distribution PaStiX is a block-oriented solver based on a supernodal approach for the factorization. The block partition of unknowns is deduced from a Nested Dissection ordering that generates recursively topological separators in the graph associated with the matrix. Then the block distribution on the processors is performed by using all the levels of parallelism exhibited by the block elimination tree.

In our case, as we use first order Nédélec edge finite elements, the unknowns are assigned to edges and we need first to build the adjacency graph associated with the matrix. It is important to notice that the recursive partitioning of this graph induces a natural splitting of the mesh elements that can be a good starting point to build a well balanced distribution of elements (tetrahedra). The first rule is to assign a tetrahedron to a processor if all of its 6 edges belong to this processor. The edge distribution based on a topological partitioning induces a strong locality at the mesh level and therefore the majority of tetrahedra falls into this category. Now considering the tetrahedra whose edges are shared between different processors, we must decide which processor owns such a tetrahedron. The proposed solution to solve this problem relies on a greedy local heuristic. As this heuristic is applied to a small number of tetrahedra, which induces a small overhead in terms of communication, the global load balance is not drastically affected. Moreover for a given number of processors, the boundary/interior edge ratio decreases with each mesh refinement reducing this communication overhead. Finally, this finite element distribution built on the coarse mesh is preserved recursively on finer meshes. Using an homogeneous and uniform refinement algorithm as described in section 3 does not modify the load balancing obtained at the first level. All these preprocessing steps are included in the geometrical setup of Algorithm 3.

Algorithm 3 Main computational kernels

1: Read coarse mesh from disk	}	Geometrical setup
2: Build the graph associated with the matrix at the coarse level		
3: Order and distribute the unknowns for the sparse direct solver		
4: Compute local meshes from unknown distribution		
5: Refine local meshes on each processing unit		
6: Save refined meshes on disk		
7: % _____	}	Numerical setup
8: Perform the assembly of the matrix at the coarse level		
9: Perform the factorization of the matrix at the coarse level		
10: % _____	}	FMG V-cycle
11: % <i>Computation of the initial guess depending on the illuminating angle</i>		
12: Perform forward-backward substitution		
13: repeat		
14: Refine the mesh		
15: Prolongate the solution		
16: until finest level		
17: % <i>Computation of the solution depending on the illuminating angle</i>		
18: repeat		
19: repeat		
20: Perform pre-smoothing steps		
21: Coarsen mesh		
22: Restrict residual		
23: until coarsest level		
24: Perform forward-backward substitution		
25: repeat		
26: Refine mesh		
27: Prolongate error		
28: Perform post-smoothing steps		
29: until finest level		
30: until convergence		

Let us notice that the geometrical setup is independent of the frequency and is performed only once as illustrated in Algorithm 1. The numerical setup including the assembly and the factorization of the matrix at the coarse level will be done for each frequency study whereas the FMG V-cycle will be done for each value of the illuminating angle.

5.2 Parallel performances

This section deals with the parallel performances of our FMG solver on a 3D realistic object. We use the TERA 100 supercomputer of CEA DAM which is composed of 4370 interconnected MESCA Bullx Série S nodes each with 4 Intel®Xeon®7500 Nehalem EX 2.27 GHz octo-core processors. The test case consists in a perfectly conducting object (looking like a “Dumb-bell”)

coated with a thick layer of material (see Figure 8). The wave number and the electromagnetic losses ensure a satisfactory convergence when using FMG as a fixed point iteration solver.

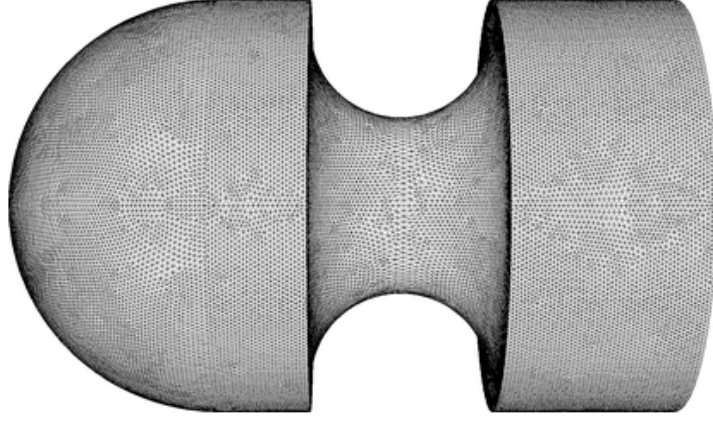


Figure 8: Mesh of the “Dumb-bell” object.

The first study compares the FMG solver with the sparse direct solver PaStiX on a small size problem for which the direct solver is still affordable on the finest mesh generated by the FMG solver regarding to memory and CPU time consumption.

We consider an initial coarse mesh of the “Dumb-bell” object with 1,288,825 dof and we apply our FMG solver with 2 refinement steps leading to a refined mesh with 21,125,245 dof. As explained earlier, PaStiX is only used on the coarse mesh. We perform 10 FMG V-cycles leading to 10^{-6} convergence accuracy in relative residual. This precision is sufficient to obtain a perfect fit between the RCS computed by the FMG solver with the one computed by the PaStiX solver on the fine mesh. Then, we can compare the achieved parallel performances with those of the direct solver PaStiX applied on this finest mesh. We use 128 cores (4 nodes) and the elapsed times (in seconds) reported in Table 2 are relative to the ordering step in the geometrical setup (Ord), to the factorization step in the numerical setup (Fact), to the forward and backward substitution steps in each V-cycle (FBsolve) and to the total elapsed time for 10 V-cycles. The last column gives the global resolution time.

Solver	Ord	Fact	FBsolve	V-cycles (10)	Global resolution
PaStiX	137	114	0.66	–	251
FMG	23	60	0.10	95	178

Table 2: Comparison between PaStiX sparse direct solver and 3-level FMG ($\nu_1 = \nu_2 = 2$) on 128 cores.

Due to smaller elapsed times for all computation steps on the coarse mesh, the global solution time of the parallel FMG solver takes 178 seconds versus 251 seconds for the parallel sparse direct solver on the fine mesh. So even on a small size problem, using our FMG solver leads to time improvement while ensuring a good accuracy for the computed solution.

The second study deals with strong scalability results of our parallel FMG solver for a medium size problem. We consider a fine mesh with 84.6 million dof obtained by two different refinement strategies. In the first one, we perform 3 refinement steps from an initial coarse mesh with 1.2 million dof and in the second one, we only perform 2 refinement steps but from a coarse mesh with 5.2 million dof. In both cases, we use $\nu_1 = \nu_2 = 2$ smoothing steps and as previously, 10 V-cycles are carried out. The number of cores used vary from 16 to 512 in the first case, and from 128 to 2048 in the second one. All timings are given in seconds.

The results are reported in Table 3 and in Table 4 respectively. The second column of the tables gives the factorization time on the coarse mesh by using PaStiX parallel sparse direct solver. The third column indicates the global time for the 10 V-cycles of our FMG parallel solver. The other columns detail the timings of the different internal steps of Algorithm 3 and they are given in percentage with regard to the 10 V-cycle total time. More precisely, in Algorithm 3, “Coarse” refers to step 21, “Refine” to step 26, “Prolong” to step 27, “FBsolve” to step 24 and “Smooth” to steps 20 and 28.

cores	Fact	V-cycles (10)	Coarse (%)	Refine (%)	Prolong (%)	FBsolve (%)	Smooth (%)
16	79.62	2285.6	10.9	40.6	0.1	0.2	48.1
32	82.72	1369.3	10.9	41.1	0.1	0.3	47.6
64	68.18	710.6	11.5	42.8	0.1	0.3	45.3
128	60.94	399.2	11.6	42.8	0.1	0.2	45.1
256	75.27	194.4	12.8	43.1	0.1	0.1	42.8
512	78.81	100.2	15.9	44.2	0.1	0.1	38.9

Table 3: Fine mesh (84.6 million dof) – coarse mesh (1.2 million dof), 4-level FMG solver.

cores	Fact	V-cycles (10)	Coarse (%)	Refine (%)	Smooth (%)
128	103.04	354.23	3.6	13.6	82.6
256	95.40	174.91	3.6	13.7	82.3
512	98.56	85.89	3.9	14.8	82.0
1024	98.96	65.88	4.2	13.5	81.0
2048	107.95	47.29	6.0	11.8	77.7

Table 4: Fine mesh (84.6 million dof) – coarse mesh (5.2 million dof), 3-level FMG solver.

As it can be seen in these two tables, the percentages of time spent in the different steps remain very similar when the number of cores is increased while the total time of the V-cycles (third column) scales very well; this highlights the very good scalability of our parallel FMG solver. At this point, we can point out that the time percentage spent in the mesh coarsening (fourth column named “Coarse”) could have been decreased in our software if we had used a hierarchical data structure representing the different mesh levels.

Nevertheless, we can remark that the factorization step does not scale because the coarse meshes lead to sparse systems which are generally too small to achieve a good efficiency on a large number of cores when using a sparse parallel direct solver. However, for our concern this point is not worrying because the RCS computation consists mainly in loops upon the direction of the illuminating wave. In these loops, only the right-hand sides of the system change while the matrix remains invariant. So the geometrical and numerical setups including the factorization step are performed only once (see Algorithm 1 and Algorithm 3) while the other computations

have to be performed for each illumination direction.

The last study deals with large size problems. Table 5 gives the scalability performances of our parallel FMG solver performing 2 refinement steps from a coarse mesh with 21.1 million dof leading to a fine mesh with 339.5 million dof. We still perform 10 V-cycles and we use from 256 to 1024 cores. Then we can notice that we have the same behaviours than the previous ones.

cores	Fact	V-cycles (10)	Coarse (%)	Refine (%)	Smooth (%)
256	117.96	628.74	3.6	13.5	82.6
512	125.56	345.00	3.6	13.5	82.3
1024	151.56	213.49	3.6	13.4	81.9

Table 5: Fine mesh (339.5 million dof) – coarse mesh (21.1 million dof), 3-level FMG.

Finally, Table 6 gives the computing times for 3 refinement steps leading to a fine mesh of 1.3 billion dof. For this very large problem, 20 V-cycles are necessary to obtain a 10^{-5} convergence accuracy in relative residual and the main computation part takes less than 1800 seconds on a moderated number of cores (1024). This parallel elapsed time is small enough to integrate the solver in a design chain to study large complex objects.

cores	Fact	V-cycles (20)	Coarse (%)	Refine (%)	Smooth (%)
1024	147.73	1715.38	4.6	17.0	78.0

Table 6: Fine mesh (1.3 billion dof) – coarse mesh (21.1 million dof), 4-level FMG.

6 Concluding remarks

By implementing a parallel FMG solver using a few levels with matrix free smoother and a robust parallel direct solver on the coarse mesh, we succeed to solve complex linear systems of 3D time harmonic Maxwell type of more than one billion unknowns. An important feature is that the final mesh, on which the Maxwell problem is solved, is generated automatically in parallel. Only a rather coarse mesh, supplied by an external mesh generator, is sufficient to describe correctly the geometry.

The considered refinement technique has many advantages beside simplicity. The most important is the following: for complex geometries the FMG is not intended to be used in all the subdomains depicted in Figure 1. Only the ones where the electromagnetic index is the largest corresponding to the smallest wave numbers (and consequently the denser meshes) are tackled with FMG. With the chosen refinement technique there is no problem of matching meshes at the interface between the subdomains of the decomposition (see again Figure 1).

Nevertheless it presents some drawbacks: if the mean size of the edges is decreased due to the creation of new edges in the interior of the existing tetrahedra, the initial edges are never divided. That is not a good feature for extremely accurate RCS calculations. Consequently, studying other refinement techniques that refine the size of all the edges deserved to be studied.

A promising new trend [23] in time harmonic Maxwell with Nédélec edge finite element discretization is the research of adaptative techniques based on a posteriori estimates of the error. That will lead to employ the process described above uniquely in the zones where it is

really necessary in a way similar to the Adaptive Mesh Refinement technique used in many simulations of nonlinear physics solved by explicit schemes.

References

- [1] ADAMS, M. AND BREZINA, M. AND HU, J. AND TUMINARO, R., *Parallel Multigrid smoothing: polynomial versus Gauss Seidel*, Journal of Computational Physics, 188 (2003), pp. 593–610.
- [2] A.H. BAKER, R.D. FALGOUT, T.V. KOLEV AND U.M. YANG, *Multigrid smoothers for ultraparallel computing*, SIAM Journal on Scientific Computing, 33 (2011), pp. 2864–2887.
- [3] I. BABUSKA, F. IHLENBURG, T. STOUBOULIS, AND S. GANGARAI, *A posteriori error estimation for finite element solution of Helmholtz equation - Part 2: Estimation of the pollution error*, International Journal for Numerical Methods in Engineering, 40(21) (1997), pp. 3883–3900.
- [4] A. BAYLISS, C. I. GOLDSTEIN, AND E. TURKEL, *On accuracy conditions for the numerical computations of waves*, Journal of Computational Physics, (1985), pp. 396–404.
- [5] J. BERENGER, *A perfectly matched layer for the absorption of electromagnetic waves*, Journal of Computational Physics, 114 (1994), pp. 185–200.
- [6] B. BERGEN, T. GRADL, U. RUEDE, AND F. HULSEMAN, *A massively parallel multigrid method for finite elements*, Computing in Science and Engineering, 8 (2006), pp. 56–62.
- [7] M. J. BOLLHOEFER, M. J. GROTE, AND O. SCHENK, *Algebraic multilevel preconditioner for the Helmholtz equation in heterogeneous media*, SIAM Journal on Scientific Computing, 31 (2009), pp. 3781–3805.
- [8] D. BOUCHE AND F. MOLINET, *Méthodes asymptotiques en Electromagnétisme*, Mathématiques et Applications, 16. Springer Verlag, 1994.
- [9] M. BREZZINA, R. FALGOUT, S. MACLACHLAN, T. MANTEUFFEL, S. MCCORMICK, AND J. RUGE, *Adaptive Algebraic Multigrid*, SIAM Journal on Scientific Computing, 27 (2006), pp. 1261–1286.
- [10] M. CHANAUD, *Un solveur linéaire haute performance de systèmes linéaires creux couplant des méthodes multigrilles et directes pour la résolution des équations de Maxwell 3D en régime harmonique*, PhD thesis, Université Bordeaux I, Talence, France, Oct. 2011.
- [11] M. CHANAUD, L. GIRAUD, D. GOUDIN, J. PESQUE, AND J. ROMAN, *A parallel full geometric multigrid solver for time harmonic maxwell problems*, Research Report RR-????, INRIA, July 2013.
- [12] E. CHOW, *An aggregation multilevel method using smooth error vectors*, SIAM Journal on Scientific Computing, 27 (2006), pp. 1727–1741.
- [13] E. CHOW, R. FALGOUT, J. HU, R. TUMINARO, AND U. YANG, *A survey of parallelization techniques for multigrid solvers*, Frontiers of Parallel Processing For Scientific Computing, (2006).

- [14] H. ELMAN, G. ERNST, AND D. O'LEARY, *A Multigrid method enhanced by Krylov subspace iteration for discrete Helmholtz equations*, SIAM Journal on Scientific Computing, 23 (2001), pp. 1291–1315.
- [15] Y. A. ERLANGGA, *Advances in iterative methods and preconditioners for the Helmholtz equation*, Archives of Computational Methods in Engineering, 15 (2008), pp. 37–66.
- [16] Y. A. ERLANGGA, C. W. OOSTERLEE, AND C. VUIK, *A novel Multigrid based preconditioner for the heterogeneous Helmholtz problems*, SIAM Journal on Scientific Computing, 27 (2006), pp. 1471–1492.
- [17] R. D. FALGOUT, *An introduction to algebraic multigrid solver*, UCRL-JRNL-220851, (2006).
- [18] R. W. FREUND AND N. M. NACHTIGAL, *Software for simplified Lanczos and QMR algorithms*, Applied Numerical Mathematics, 19 (1995), pp. 319–341.
- [19] M. J. GANDER, *Classical multigrid method: Efficient for diffusion problems but ineffective for wave propagation problems*, in Conf. at Inria Sophia Antipolis, July 2009.
- [20] J. GOPALAKRISHNAN, J. PASCIAK, AND L. DEMKOWICZ, *Analysis of a Multigrid algorithm for time harmonic Maxwell equations*, SIAM Journal on Scientific Computing, 42 (2004), pp. 90–108.
- [21] T. GRADL AND U. RUEDE, *High performance multigrid in current large scale parallel computers*, in 9th Workshop on Parallel Systems and Algorithms, Dresden, 2008.
- [22] E. HABER AND S. MACLACHLAN, *A fast method for the solution of the Helmholtz equation*, Journal of Computational Physics, 230 (2011), pp. 4403–4418.
- [23] D. HARUTYUNYAN, F. IZSAK, J. VAN DER VEGT, AND M. BOTCHEV, *Adaptive finite element techniques for the Maxwell equations using implicit a posteriori error estimates*, Computer Methods in Applied Mechanics and Engineering, 197 (2008), pp. 1620–1638.
- [24] P. HENON, P. RAMET, AND J. ROMAN, *PaStiX a high performance parallel direct solver for sparse symmetric definite system*, Parallel Computing, 28 (2002), pp. 301–321. <http://pastix.gforge.inria.fr>.
- [25] R. HIPTMAIR AND J. XU, *Nodal auxiliary space preconditioning in $H(\text{curl})$ and $H(\text{div})$ spaces*, SIAM Journal on Numerical Analysis, 45 (2007), pp. 2483–2509.
- [26] G. HOUZEAUX, R. RAUL DE LA CRUZ, H. OWEN, AND M. VASQUEZ, *Parallel uniform mesh multiplication applied to Navier-Stokes solver*, Computer and Fluids, <http://dx.doi.org/10.1016/j.compfluid.2012.04.017> (2012).
- [27] J. HU, R. TUMINARO, P. BOCHEV, C. J. GARASI, AND A. ROBINSON, *Toward an h -independent Algebraic Multigrid for Maxwell's equation*, SIAM Journal on Scientific Computing, 27 (2006), pp. 1669–1688.
- [28] J. JONES AND B. LEE, *A Multigrid method for variable coefficient Maxwell's equation*, SIAM Journal on Scientific Computing, 27 (2006), pp. 1689–1708.
- [29] S. KIM AND S. KIM, *Multigrid simulation for high-frequency solution of the Helmholtz problem in heterogeneous media*, SIAM Journal on Scientific Computing, 24 (2002), pp. 684–701.

- [30] T. KOLEV AND P. VASSILEVSKI, *Parallel Auxiliary Space AMG for $H(\text{curl})$ Problems*, J. Comput. Math, 27 (2009), pp. 604–623.
- [31] B. LEE, T. MANTEUFFEL, S. F. MCCORMICK, AND J. RUGE, *First order system least-squares for the Helmholtz equation*, SIAM Journal on Scientific Computing, 21 (2000), pp. 1228–1251.
- [32] I. LIVSHITS AND A. BRANDT, *Accuracy properties of the wave ray Multigrid algorithm for Helmholtz equation*, SIAM Journal on Scientific Computing, 28 (2006), pp. 1228–1251.
- [33] S. MACLACHLAN AND C. W. OOSTERLEE, *Algebraic multigrid solvers for complex-valued matrices*, SIAM Journal on Scientific Computing, 30 (2008), pp. 1548–1571.
- [34] S. MACLACHLAN AND Y. SAAD, *A greedy strategy for coarse grid selection*, SIAM Journal on Scientific Computing, 29 (2007), pp. 1825–1853.
- [35] M. MAGOLU MONGA MADE, R. BEAUWENS, AND G. WARZÉE, *Preconditioning of discrete Helmholtz operators perturbed by a diagonal complex matrix*, Communications in numerical methods in engineering, 16 (2000), pp. 801–817.
- [36] K. MER, A. BACHELOT, AND E. DARRIGRAND, *Coupling of a Multilevel Fast Multipole method and a Microlocal discretization for the 3D Integral Equation of Electromagnetism*, C.R. Acad. Sci. Maths, 336 (2003), pp. 505–510.
- [37] L. N. OLSON, J. SCHOEDER, R. TUMINARO, AND B. JACOB, *A general interpolation strategy for algebraic multigrid using energy minimization*, SIAM Journal on Scientific Computing, 33 (2011), pp. 966–991.
- [38] B. REPS, W. VANROOSE, AND H. BIN ZUBAIR, *On the indefinite Helmholtz equation: complex stretched absorbing boundary layers, iterative analysis and preconditioning*, Journal of Computational Physics, 229 (2010), pp. 8384–8405.
- [39] C. D. RIYANTI, A. KONONOV, Y. ERLANGGA, C. VUIK, C. OSTERLEE, R. PLESSIX, AND W. MULDER, *A parallel multigrid based preconditioner for the 3D heterogeneous high-frequency Helmholtz equation*, Journal of Computational Physics, 224 (2007), pp. 431–448.
- [40] Y. SAAD AND M. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM Journal on scientific and statistical computing, 7 (1986), pp. 856–869.
- [41] M. B. VAN GIJZEN, Y. A. ERLANGGA, AND C. VUIK, *Spectral analysis of the discrete Helmholtz operator preconditioned with a shifted Laplacian*, SIAM Journal on Scientific Computing, 29 (2007), pp. 1942–1958.



**RESEARCH CENTRE
BORDEAUX – SUD-OUEST**

200 avenue de la Vielle Tour
33405 Talence Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399